

Amendments To Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

Listing of Claims:

1. (Currently Amended) A method comprising:

receiving, via radio frequency (RF) communication, information identifying a currently running software and information related to a staged software to be loaded on a restart of a software-defined radio device;

transferring, via radio frequency (RF) said RF communication and in response to said staged software being improper, software directly to [[a]] said software-defined radio device from a software server to create transferred software, said software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said transferred software is stored in at least a portion of a data store associated with said software-defined radio device;

transferring, via said RF communication, a back-up copy of said transferred software that is executed in response to runtime errors, wherein said runtime errors are generated by executing said transferred software; and

sending an instruction, via said RF communication, directly to said software-defined radio device identifying said transferred software or said software currently running in said software-defined radio device as a ~~selected software application~~ said staged software to be loaded by said software-defined radio device in response to a restart of said software-defined radio device;

— automatically switching from said ~~selected software application~~ to a different version of said ~~selected software application~~ in response to a first fault, without uninstalling said ~~selected software application~~; and

uninstalling said ~~selected software application~~ and reverting back to said different version of said ~~selected software application~~ in response to a second fault.

2. (Cancelled)

3. (Currently Amended) The method according to claim 1, further comprising monitoring said transferring of said transferred software and monitoring said loading of said ~~selected software application~~ staged software.

4 – 5. (Cancelled)

6. (Previously Presented) The method according to claim 1, wherein said instruction identifies a software version.

7. (Previously Presented) The method according to claim 1, wherein said software-defined radio device comprises a plurality of software defined radio devices.

8. (Currently Amended) The method according to claim 1, further comprising receiving an error indication in response to said first fault or said second fault being detected in at least one of said transferring of said transferred software or said loading of said ~~selected software application~~ staged software.

9. (Original) The method according to claim 1, wherein said transferred software comprises a plurality of software components.

10. (Cancelled)

11. (Previously Presented) The method according to claim 1, further comprising receiving a software listing from said software-defined radio device, said software listing identifying software currently available on said data store.

12. (Previously Presented) The method according to claim 1, wherein said transferred software is stored in a second data store associated with said software-defined device.

13. (Previously Presented) The method according to claim 12, wherein said second data store is nonvolatile.

14. (Previously Presented) The method according to claim 1, wherein said transferring of said transferred software occurs in response to said software-defined radio device continuing to perform software-defined radio functions.

15. (Previously Presented) The method according to claim 1, wherein said software server comprises a computer operatively connected to said software-defined radio device via a wireless communications network.

16. (Currently Amended) A method comprising:

transmitting, via radio frequency (RF) communication, information identifying a currently running software and information related to a staged software to be loaded on a restart of a software-defined radio device;

receiving, via ~~radio frequency (RF)~~ said RF communication directly from a software server and in response to said staged software being improper, transferred software at a software-defined radio device, said software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said software currently running in said software-defined radio device is stored in a first non-volatile data store area;

receiving, via said RF communication, a back-up copy of said transferred software that is executed in response to runtime errors, wherein said runtime errors are generated by executing said transferred software;

storing said transferred software in a second non-volatile data store area distinct from said first non-volatile data store area;

receiving, via said RF communication directly from said software server, an instruction at said software-defined radio device identifying said transferred software or said software currently running in said software-defined radio device as ~~a selected software application~~ said staged software to be loaded by said software-defined radio device in response to a restart of said software-defined radio device;

providing an error indication in response to a first fault detection, and selecting a different software version of said selected software application based on a particular error in said error indication; ~~without uninstalling said selected software application;~~

~~uninstalling said selected software and reverting back to said different software version of said selected software application in response to a second fault detection;~~

responsive to a restart instruction, restarting said software-defined radio device and loading said ~~selected software application~~ staged software; and

verifying said selected software application is loaded successfully.

17. (Currently Amended) The method according to claim 16, further comprising automatically reverting from said ~~selected software application~~ staged software to a different software version without an instruction from said software server, wherein said automatically reverting is in response to said ~~selected software application~~ staged software encountering an error which causes said software-defined radio device to stop functioning properly.

18. (Cancelled)

19. (Previously Presented) The method according to claim 16, further comprising: monitoring said receiving transferred software step; and wherein said fault detection is detected in said receiving transferred software step.

20. (Cancelled)

21. (Currently Amended) The method according to claim 16, wherein said ~~selected software application~~ staged software identifies a software version.

22. (Previously Presented) The method according to claim 16, further comprising providing a software listing to a remote location, said software listing identifying software currently available on said first non-volatile data store area.

23 - 24. (Cancelled)

25. (Previously Presented) The method according to claim 16, further comprising in response to receipt of said transferred software, decompressing said transferred software.

26. (Previously Presented) The method according to claim 16, wherein receiving said transferred software occurs while said software-defined radio device continues to perform software-defined radio functions.

27. (Currently Amended) A software-defined radio device comprising:

a radio frequency (RF) communications interface configured to transmit information identifying a currently running software and information related to a staged software to be loaded on a restart and receive transferred software and an instruction directly from a software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software configured to be currently running in said software-defined radio device and in response to said staged software being improper, and wherein said software server comprises a man-machine interface configured to receive said instruction comprising a selected software configured to specify whether said transferred software or said software configured to be currently running in said software-defined radio device will be loaded in response to a restart of said software-defined radio device, wherein said RF communications interface is further configured to receive a back-up copy of said transferred software that is configured to be executed in response to runtime errors, wherein said runtime errors are generated by executing said transferred software;

a data store associated with said software-defined radio device configured to store said transferred software in at least a portion of said data store; and

a processor programmed to:

load said selected software to said software-defined radio device in response to said restart of said software defined radio device; and

automatically revert, without an instruction from said software server, from said selected software to a different software version responsive to said selected software encountering a first fault, ~~without uninstalling said selected software~~; and

~~uninstall said selected software and revert back to said different software version~~
~~in response to a second fault.~~

28. (Previously Presented) The device according to claim 27, wherein said processor is further programmed to determine that said software and said instruction are received successfully and to determine that said selected software is loaded successfully.

29. (Cancelled)

30. (Previously Presented) The device according to claim 27, wherein said processor is further programmed to decompress said transferred software, and wherein said software server further comprises a compression application for compressing said software prior to said software being transferred.

31. (Previously Presented) The device according to claim 27, wherein said transferred software comprises a plurality of software components.

32. (Cancelled)

33. (Previously Presented) The device according to claim 27, wherein said RF communications interface is further configured to transmit a software listing identifying software currently available on said data store to said software server.

34. (Previously Presented) The device according to claim 27, further comprising a second data store associated with said software-defined device configured to store said transferred software.

35. (Previously Presented) The device according to claim 34, wherein said second data store is nonvolatile.

36. (Previously Presented) The device according to claim 27, wherein said processor is further programmed to receive said software from said software server while said software-defined radio device performs software-defined radio functions.

37-38. (Cancelled)

39. (Currently Amended) A tangible, non-transitory computer-readable medium having stored thereon [[.]] instructions ~~computer-executable instructions that, in response to execution by a computing device, cause said computing device to perform a method comprising:~~

instructions for transmitting, via radio frequency (RF) communication, information identifying a currently running software and information related to a staged software to be loaded on a restart of a software-defined radio device;

instructions for receiving, via radio frequency (RF) RF communication direct from a software server and in response to said staged software being improper, transferred software at [[a]] said software-defined radio device, said software server remotely located with respect to said software-defined radio device, wherein said transferred software is another version of software currently running in said software-defined radio device, and wherein said software currently running in said software-defined radio device is stored in a first data store area;

instructions for receiving, via said RF communication, a back-up copy of said transferred software that is executed in response to runtime errors generated by executing said transferred software;

instructions for storing said transferred software in a second data store area distinct from said first data store area;

instructions for receiving via said RF communication direct from a software server an identification of said transferred software or said currently running software as a selected software application said staged software to be loaded by said software-defined radio device in response to a restart of said software-defined radio device;

instructions for providing an error indication in response to a first fault detection, and selecting a different software version of said selected software application staged software based on a particular error in said error indication, ~~without uninstalling said selected software application;~~

~~uninstalling said selected software and reverting back to said different software version of said selected software application in response to a second fault detection;~~

instructions for restarting said software-defined radio device and loading said ~~selected software~~ staged software, responsive to a restart instruction, while maintaining said software currently running in said software-defined radio device in said first data store area and said transferred software in said second data store area; and

instructions for verifying said ~~selected software application~~ staged software is loaded successfully.